2 of 3

8-P \$0.50

AN APPLICATION OF DYNAMIC PROGRAMMING TO THE COLORING OF MAPS

Richard Bellman

May 1964

P-2918

SUMMARY

In some previous papers and books, we have indicated some applications of dynamic programming to some puzzles and popular diversions. Here we wish to consider some aspects of map coloring. We shall both extend and invert the famous four-color problem.

The method we present is a combination of exact and heuristic techniques. Conceivably, it may offer an approach to some general theoretical results. At the moment, it is designed to resolve the problem of how a particular map is to be colored with three or four colors if one actually wants to color it. In general, the computational procedure given requires a digital computer.

This paper was prepared for publication in the Mathematics Magazine.

AN APPLICATION OF DYNAMIC PROGRAMMING TO THE COLORING OF MAPS

1. INTRODUCTION

In some previous papers and books [1,2,3,4,5,6,7], we have indicated some applications of dynamic programming to some puzzles and popular diversions. Here we wish to consider some aspects of map coloring. We shall both extend and invert the famous four-color problem.

The method we present is a combination of exact and heuristic techniques. Conceivably, it may offer an approach to some general theoretical results. At the moment, it is designed to resolve the problem of how a particular map is to be colored with three or four colors if one actually wants to color it.

2. SEPARATING BOUNDARIES

To illustrate our procedure, let us consider the following region (see Fig. 1).

The heavy lines are called separating boundaries. They have the property that the subregions contained between the i-th and $(i + 1)$-st separating boundaries separate the region into two parts which have no common boundary.

3. EXTENDED MAP COLORING PROBLEMS

In what follows, we shall assume that we are allowed k distinct colors, where $k = 3$ or 4. The case

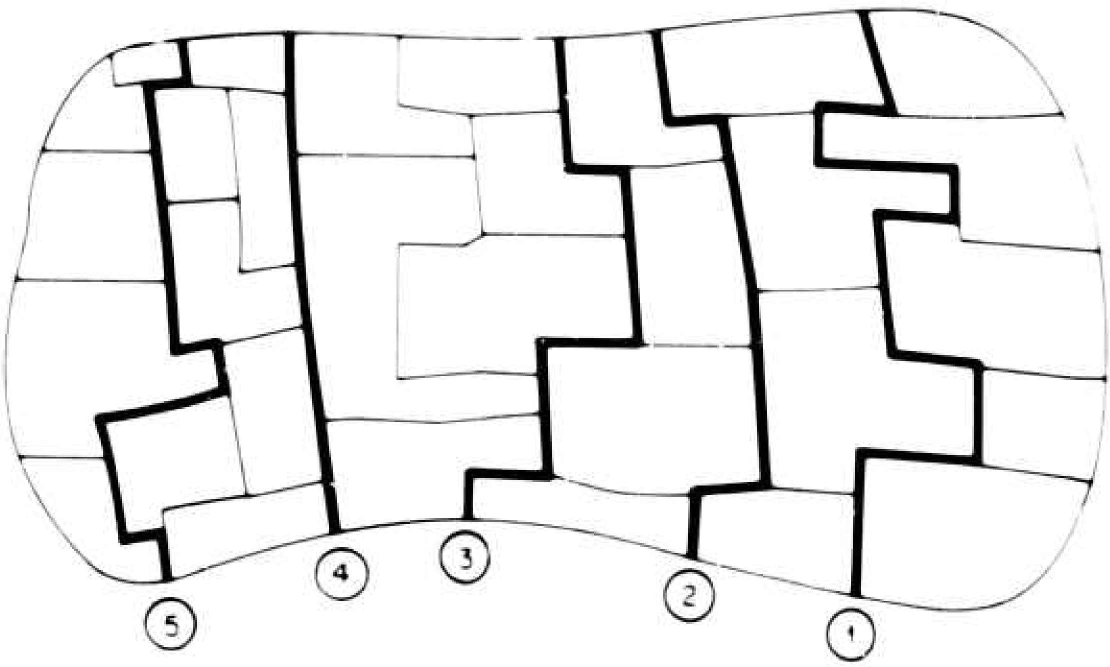

Fig. 1

$k = 2$ is uninteresting, and we tacitly assume that the four-color conjecture is true, so that there is no point using $k \geq 5$.

Suppose the region to the right of the first separating boundary, $\overset{\frown}{\underset{\smile}{1}}$, has been colored in some fashion, which is to say a color has been assigned to each state. Let us now define what we mean by <u>overlap</u>. By an overlap we mean that two states with a common boundary have the same color.

Given the initial coloring, C, with its possible overlaps, we want to color the remaining states so as to minimize the additional overlaps. These overlaps occur both inside the region to the left at $\overset{\frown}{\underset{\smile}{1}}$ and at the boundary, $\overset{\frown}{\underset{\smile}{1}}$.

To treat this problem, we use an imbedding technique. We suppose that the region to the right of the region between the i-th and $(i - 1)$-st separating boundaries is colored with a color scheme C, and ask for the coloring to the left of $\overset{\frown}{\underset{\smile}{i}}$ which will minimize the additional overlaps incurred.

Let, for $i = 1,2,3,4,5$,

(3.1) $f_i(C)$ = the number of overlaps incurred coloring the region to the left of $\overset{\frown}{\underset{\smile}{i}}$ using an optimal coloring scheme, given that the strip between $\overset{\frown}{\underset{\smile}{i-1}}$ and $\overset{\frown}{\underset{\smile}{i}}$ is colored with C.

The best color scheme for the original scheme is determined by minimizing $f_1(C)$ over C. This is to be carried out by straightforward enumeration. If the separating boundaries are chosen judiciously, we can actually determine this number computationally.

4. BASIC RECURRENCE RELATION

Let

(4.1) $P_i(C,C')$ = number of overlaps resulting from a coloring scheme C' in the region between i and $i+1$ when the region between i and $i-1$ has the scheme C.

Then the principle of optimality [1], [2] yields the recurrence relation

$$f_i(C) = \min_{C'} [P_i(C,C') + f_{i+1}(C')], \quad i = 1,2,3,4, \tag{4.2}$$

with

$$f_5(C) = \min_{C'} [P_5(C,C')]. \tag{4.3}$$

5. DISCUSSION

The numerical utility of the foregoing approach depends upon the number of possible coloring schemes associated with each region. At present, the larger

digital computers will allow about 3×10^4 values without any difficulty, and, with a little juggling, 5×10^4, can be accomodated; see [2] for detailed discussion. Within about a year, this will be increased to about 2×10^5, and possibly 4×10^5.

A subregion with s states can be colored in k^s ways, allowing all possible schemes involving k colors. If $k = 4$, $s = 10$, we have $4^{10} \simeq 10^6$ possibilities. Using some discretion in our choice of coloring schemes, we can reduce this greatly.

The same method can be used to treat more general problems where we assign a number of different properties to each state and desire a minimum number of partial overlaps. This brings us to questions of Euler squares and experimental design, which we shall discuss elsewhere.

Finally, we can in the foregoing way obtain a recurrence relation for the number of coloring schemes involving at most m overlaps, and so on.

REFERENCES

1. Bellman, R., Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1957.

2. Bellman, R., and S. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1962.

3. Bellman, R., "Sequential Machines, Ambiguity and Dynamic Programming," J. Assoc. Comput. Machinery, Vol. 7, 1960, pp. 24–28.

4. ———, "Dynamic Programming and Difficult Crossing Puzzles," Math. Mag., January–February 1962, pp. 27–29.

5. ———, "On Some Mathematical Recreations," Amer. Math. Monthly, Vol. 69, 1962, pp. 640–643.

6. ———, "Dynamic Programming Treatment of the Traveling Salesman Problem," J. Assoc. Comput. Machinery, Vol. 9, 1962, pp. 61–63.

7. ———, Dynamic Programming and Markovian Decision Processes with Particular Application to Baseball and Chess, to appear.